

Pythongrundlagen

# text	Kommentar
type(Objekt)	Typ des Objektes
print(Objekt)	Text in der Konsole ausgeben
len(x)	Anzahl der Elemente in 'x'
list.append(x)	'x' in Liste 'list' hinzufügen
list.remove(x)	Erstes 'x' aus Liste 'list' löschen
a.copy()	Kopie eines Arrays oder einer Liste 'a'
fun=lambda x: Anweisung	Einzeilige Funktion erstellen
while Bedingung:	while-Schleife
for a in b:	for-Schleife
f=open('a'), f.close()	Datei 'a' öffnen/schließen
f.readlines()	Alle Zeilen einer Datei lesen
def fun(a):	Funktionsdefinition (mit einem Parameter 'a')
return x	Rückgabewert einer Funktion
if, elif, else	Bedingte Anweisungen
break, continue	Schleife abbrechen bzw. von oben weitermachen
'String'.format(x)	Stringmanipulation
a[2:7]	Slicing
range(5)	Iterierbares Objekt
*a	Entpackt die Liste 'a'
try & except	Fehler abfangen
assert bedingung, text	Bedingung überprüfen, sonst Fehlermeldung
class Name(Vater):	Python-Klasse 'Name' (erbt von 'Vater')
__init__(self):	Initialisierungsmethode einer Klasse
isinstance(x,y)	Überprüft ob 'x' den Klassentyp 'y' hat
issubclass(x,y)	Überprüft ob 'x' Unterklasse von 'y' ist

NumPy (import numpy as np)

np.cos(x), np.sin(x), np.exp(x), np.sqrt(x)	Cosinus, Sinus, Exponentialfunktion, Wurzel
np.sum(x), np.prod(x)	Summe, Produkt von 'x'
np.loadtxt('a')	Textdatei (mit Zahlen) einlesen
np.array(a)	Liste/Tupel in array umwandeln
np.linspace(a,b)	Äquidistant verteilte Punkte in [a,b]
np.random	Modul für (Pseudo)Zufallszahlen
np.linalg.det(A)	Determinante von 'A'
np.linalg.norm(a)	Norm von Matrizen/Vektoren
np.linalg.solve(A,b)	Lineares Gleichungssystem lösen
a.astype('x')	Kopie des Arrays 'a' eines bestimmten Typs
np.arange(x)	np.array(range(x))
A.flatten()	Eindimensionale Version des Arrays 'A'
np.tril/triu(A)	Unteres/Oberes Dreieck der Matrix 'A'
np.kron(a,b)	Kronecker-Symbol
np.diag(a)	Diagonalmatrix erstellen/Diagonale rausziehen
np.min/np.max(a)	Minimales/Maximales Element von 'a'
np.argmax/argmin(a)	Index mit dem größten/kleinsten Eintrag von 'a'
np.ones(x), np.zeros(x)	Arrays bestimmter Dimension mit Einsen/Nullen
np.eye(n)	Einheitsmatrix der Größe n x n
np.vstack(tuple), concatenate(tuple)	Arrays aus 'tuple' aneinanderpacken
np.linalg.svd(A)	Singulärwertzerlegung von 'A'
np.linalg.eig(A)	Eigenwerte und -vektoren von 'A'
np.linalg.qr(A)	QR-Zerlegung von 'A'
scipy.linalg.lu(A)	LR-Zerlegung von 'A' (vorher import scipy.linalg)
a.size	Anzahl Elemente des Arrays 'a'
a.shape	Dimensionen des Arrays 'a'
a.ndim	Anzahl der Dimensionen des Arrays 'a'
A.T	'A' transponiert
A.conj()	'A' komplex konjugiert (komponentenweise)
np.reshape(a,...)	Form des Arrays 'a' ändern
np.logical_and/or(a,b)	Komponentenweises UND/ODER auf Arrays 'a' und 'b'
np.clip(a,b,c)	Werte von 'a' auf das Intervall[b,c] beschränken

Matplotlib (import matplotlib.pyplot as plt)

<code>plt.figure()</code>	Neues Plotfenster
<code>plt.plot(x,f(x))</code>	Plottet 'f(x)' im Intervall 'x'
<code>plt.title('text')</code>	Titel des Plots
<code>plt.legend(tuple)</code>	Legende des Plots
<code>plt.subplot(a,b,c)</code>	Unterplots im Plotfenster
<code>plt.xlabel('a')</code>	x/y-Achse beschriften
<code>plt.semilogx/y(), loglog()</code>	Logarithmische Achsen im Plot
<code>plt.axis(a)</code>	Achsenkalierung ändern
<code>plt.xlim(tuple)</code>	Achsen Grenzen setzen
<code>plt.imread('name')</code>	Liest ein Bild ein
<code>plt.imshow('name',bild)</code>	Speichert ein Bild
<code>plt.imshow(bild)</code>	Zeigt 'bild' an
<code>plt.savefig('name')</code>	Speichert einen Plot
<code>fig.savefig('name')</code>	Speichert den Plot 'fig'